# From Access and Integration to Mining of Secure Genomic Data Sets across the Grid

**Professor Richard O. Sinnott,**
**National e-Science Centre**
**University of Glasgow**
**Glasgow G12 8QQ**
**ros@dcs.gla.ac.uk**

**Abstract**

The UK Department of Trade and Industry (DTI) funded BRIDGES project (Biomedical Research Informatics Delivered by Grid Enabled Services) has developed a Grid infrastructure to support cardiovascular research. This includes the provision of a compute Grid *and* a data Grid infrastructure with security at its heart. In this paper we focus on the BRIDGES data Grid. A primary aim of the BRIDGES data Grid is to help control the complexity in access to and integration of a myriad of genomic data sets through simple Grid based tools. We outline these tools, how they are delivered to the end user scientists. We also describe how these tools are to be extended in the BBSRC funded Grid Enabled Microarray Expression Profile Search (GEMEPS) to support a richer vocabulary of search capabilities to support mining of microarray data sets. As with BRIDGES, fine grain Grid security underpins GEMEPS.

**Keywords**: Grid, Security, microarray data;

## 1. Introduction

The completion of sequencing of human and several other eukaryotic genomes as well as more than a hundred microbial ones marks the beginning of the post-genomic era, where emphasis has shifted from searching for genes onto understanding of their function. The new discipline - functional genomics - was born and its success has been grossly facilitated by development of modern post-genomic technologies that enabled comprehensive studies of mRNA, protein and metabolite complements of biological samples. Due to the high-throughput fashion of these technologies, functional genomics has been generating large amounts of data. To be able to sensibly analyse these data well-designed data standards are required. Human Proteome Organisation's (HUPO) Proteomics Standards Initiative (PSI) [1] has adopted the PEDRO (Proteomics Experiment Data Repository) standard [2] for proteomic data. Recently, the ArMet (Architecture for Metabolomics) model was proposed for metabolomic data [3]. The most advanced work, however, has been done by the microarray community through the development of the MIAME (Minimum Information about a Microarray Experiment) standard for transcriptomic data [4]. These days leading journals require microarray data associated with publications to be MIAME compliant, and this standard has been adopted by several public data repositories.

Data stored in these repositories can be easily searched using various terms belonging to carefully crafted controlled vocabularies. However, none of the existing repositories provides means for searching the deposited data by the results of a particular microarray experiment. In other words, currently a researcher cannot assess if a similar experiment has been undertaken previously; if other experiments have produced similar results, or generally understand how their experiment compares to previously undertaken experiments. More generally from a biological perspective we can introduce the concept of a Simple FunctIonal geNomics eXperiment (SFINX) where SFINX represents a comparison of two biological conditions represented by two groups of biologically replicated samples (healthy vs. diseased tissue, wild type vs. mutant animals, drug-treated vs. control cells, etc.). Each sample contains a population of elements (e.g. mRNAs, proteins, metabolites). By identifying statistically significant differences in the above populations we want to learn about a biological process that could explain a difference between the two conditions. Modern post-genomic technologies enable quantitative measurement of changes in populations of elements. In case of gene expression arrays that measure differences in the transcriptome content, the elements are genes (or rather mRNAs). In the case of quantitative proteomics technologies such as

Differential Gel Electrophoresis [46] or Isotope Coded Affinity Tags [47] that measure differences in proteome composition the elements are proteins. In the case of quantitative metabolomic technologies such as Fourier Transform Ion Cyclotron Mass Spectrometry [48] that measure differences in metabolite concentrations the elements are metabolites. As a result of a SFINX, each particular element is given a measure of its change and the complete list of these measures constitutes a profile that fully characterises the experiment. It is perfectly reasonable that after calculating a SFINX profile a researcher would like to know if somebody somewhere performed another experiment of a similar profile. Such information could lead to the assumption that at the first approximation similar biological processes took place in both experiments. This could potentially save time, efforts and resources in identifying such processes. Unfortunately, at present there are no mechanisms for such search available in the public repositories of functional genomics' data.

In order to make such searches meaningful, several conditions have to be fulfilled. Firstly, the SFINX profile has to be reliable; secondly it has to have a set of sub-profiles corresponding to different level of confidence; thirdly the library of profiles has to be constructed using a standardised method; and lastly a similarity measure between profiles has to be established. The Sir Henry Wellcome Functional Genomics Facility at the University of Glasgow (SHWFGF) have developed a number of new techniques to analyse large genomic datasets such as microarray results. These techniques try to combine statistical and biological reasoning in an automated framework. They include: *Rank Products* (RP) - a powerful test statistics to detect differentially regulated genes in replicated experiments [49]; *iterative Group Analysis* (iGA) - an algorithm to compute physiological interpretations of experiments based on existing functional annotations [50], and *Graph based interative Group Analysis* (GiGA) - a graph-based extension of iGA that uses expression data to highlight relevant areas in an "evidence network" to facilitate data interpretation and visualisation [51]. These methods have shown how with *local data sets*, a novel, fully automated pipeline for the analysis of Affymetrix GeneChip arrays can be supported. This schema has been running successfully for several months in the SHWFGF. So far analysis of nearly 500 SFINXs that comprised nearly 2000 chips has been performed.

To extend this approach to deal with *distributed data sets* requires several challenges to be overcome. Firstly, data must be found and accessed – often requiring local security issues to be dealt with. Secondly it must be integrated with other data sets – where remote data sets are continually evolving. Thirdly and ideally it should be mined to bring more understanding and support richer mechanisms for comparison and evaluation of biological experiments/results. The BRIDGES project [14] has developed a Grid infrastructure that addresses the first two of these concerns: data access and integration. A follow on BBSRC funded project Grid Enabled Microarray Expression Profile Search (GEMEPS) [52] will enhance this infrastructure to move towards data mining capabilities.

Grid technologies directly address many of the difficulties present in large scale heterogeneous distributed systems where collections of remote data and compute resources are to be seamlessly accessed and used. One of the primary challenges that Grid technologies face is managing the access to and usage of a broad array of remote, evolving and largely autonomous data sets (in the sense of being managed by separate organisations). Whilst it is possible to have data curation centres where for example microarray data are stored and maintained centrally, e.g. such as the Medical Research Council/Imperial College funded Clinical Science Centre microarray data warehouse [5]; large centralised centres are costly to set up and subsequently manage and have a significant drawback, namely they require that scientists are prepared to hand over their data sets to a third party to manage and ensure that appropriate security mechanisms are in place. Scientists are generally unwilling to make their microarray data sets (or research data sets more generally) before their experiments are formally published in journals / conferences [6]. As such, these data curation repositories are always likely to be populated with older data sets, thus scientists wishing to perform experiments are unable to determine whether recent experiments have been performed already and hence unable to perform any useful comparison until papers have been published. This can, depending upon the journal, be especially time consuming.

A better model is to allow scientists to keep and maintain their own local data sets, and allow them to provide secure access to their data in a tightly controlled setting, e.g. to specific colleagues or centres wishing to compare mutually beneficial experiments. To achieve this and bearing in mind the competitive

nature of research and costs incurred in running experiments, security of data is an important factor. Individual sites will of course have their own procedures and policies for how they deal with data security, however the Grid community has developed generic security solutions that can be applied to augment existing security infrastructures. Through these additional mechanisms local security policies can be enforced restricting and controlling access to research data sets that might otherwise not be available, i.e. not yet published data. This is achievable through recent Grid security standardisation activities [7], recent technological developments [8,9,10,11,12,13] and direct experiences of the National e-Science Centre (NeSC) at the University of Glasgow in projects such as the JISC funded DyVOSE project [15], the MRC funded VOTES project [16] and the CSO funded Genetics and Healthcare project [17].

## 2. BRIDGES Project Overview

Arguably the primary objective in applying Grid technology is to establish virtual organisations (VOs). VOs allow shared use of computational and data resources by collaborating institutions/scientists. Establishing a VO requires that efficient security access control mechanisms to the shared resources by known individuals are in place. One example of a VO is the Wellcome Trust funded (£4.34M) Cardiovascular Functional Genomics (CFG) project [18] who are investigating possible genetic causes of hypertension, one of the main causes of cardiovascular mortality. This consortium which involves five UK sites and one Dutch site is pursuing a strategy combining studies on rodent models of disease (mouse and rat) contemporaneously with studies of patients and population DNA collections. The BRIDGES project has been funded by the UK Department of Trade and Industry to develop a computational infrastructure to support the needs of CFG.

Currently many of the activities that the CFG scientists undertake in performing their research are done in a time consuming and largely non-automated manner. This is typified through “internet hopping” between numerous life science data sources. For example, a scientist might run a microarray experiment and identify a gene (or more likely set of genes) being differentially expressed. This gene is then used as the basis for querying a remote data source (e.g. MGI in Jackson [29]). Information retrieved from this query might include a link to another remote data source, e.g. on who has published a paper on this particular gene in MedLine [30] or PubMed [31]. Information from these repositories might include links to ensembl [32] where further information on the gene, e.g. its start/end position in a given chromosome can be established. Such sequences of navigations typify the research undertaken by scientists.

A key component of the BRIDGES architecture is the Data Hub (Figure 1). This represents both a local data repository, together with data made available via externally linked Grid accessible data sets. These data sets exist in different heterogeneous, remote locations with differing security requirements. Some data resources are held publicly (e.g. genome databases such as Ensembl [32], gene function databases such as OMIM [35] and relevant publications databases such as MedLine [30]); whilst others are for usage only by specific CFG project partners (e.g. quantitative trait loci (QTL) data sets [36]).

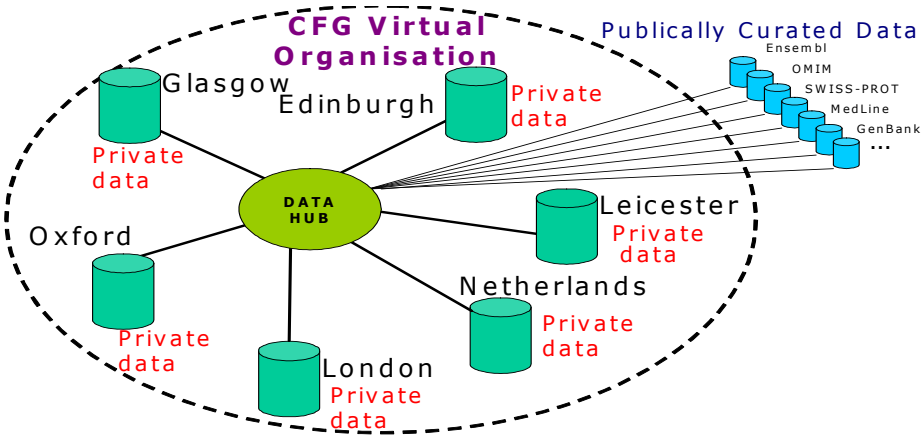


Figure 1: Data Distribution and Security of CFG Partners

Currently the public data sets are accessible via two different technologies: IBM's Information Integrator (IBM-II) – formerly known as DiscoveryLink [33], and the Open Grid Service Architecture – Data Access and Integration (OGSA-DAI) technology [34]. IBM-II technology has been developed to meet the challenge of integrating and analyzing large quantities of diverse scientific data from a variety of life sciences domains offering single-query access to existing databases, applications and search engines. This is achieved through wrappers which use the data source's own client-server mechanism to interact with the sources in their native dialect. Through IBM-II access to a broad array of heterogeneous data sets can be achieved, e.g. relational databases, XML databases, Excel spreadsheets, flat files etc. In a similar vein, the OGSA-DAI technology provides a generic data access and integration mechanism overcoming issues related to the heterogeneity of technologies and data sets as well as the remoteness of data sets themselves. This technology is being continually refined and extended to be compliant with on-going Grid standardisation efforts.

To support the life science community it is essential that applications are developed that allow them simplified access to life science data sets as well as to personalise their environments. The personalisation might well include the data sources that are of most interest to the scientists and the information that they are most interested in from those data sources.

The BRIDGES project has developed two client side tools which act as front ends to this data hub: MagnaVista and GeneVista.

## 2.1 MagnaVista

The Java application MagnaVista provides a single, personalisable way in which a broad range of genomic data sets can be simultaneously accessed and used based on querying over gene names (or synonyms) – which themselves may have originated from particular local microarray experiments. This application provides a completely configurable environment through which the scientists can navigate to and access a broad array of life science data sets of relevance to their research. The basic user interface to MagnaVista is depicted in Figure 2.

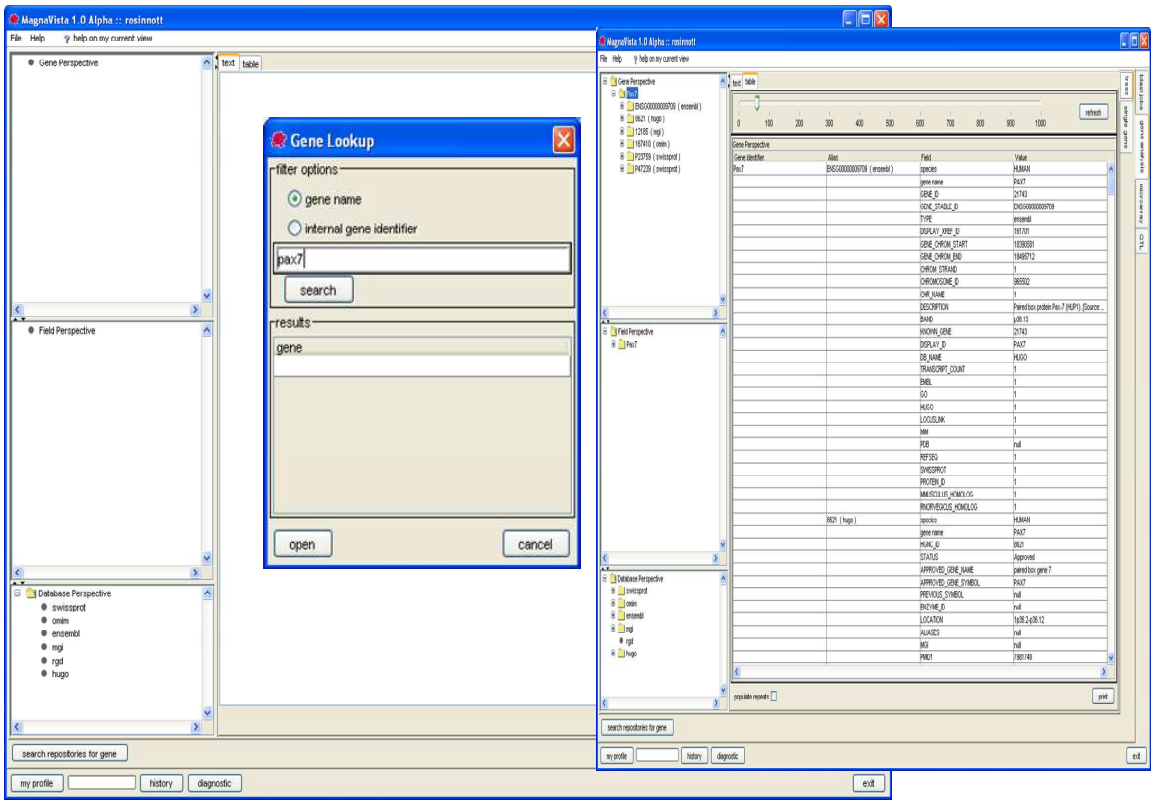


Figure 2: MagnaVista Basic Usage for Gene Query

Here the user can include the genes that they are most interested in (central pop up window). The lower left corner of Figure 2 lists the remote data sources that are accessible (SWISS-PROT [37], MGI [29], Ensembl [32] (rat, mouse, human DBs), RGD [38], OMIM [35]). The pop up window to the right of Figure 2 shows the data sets that are returned to the user upon submission of the query.

Thus rather than the user manually hopping to each of these remote resources, a single query is used to deliver collections of data associated with the genes of interest. To support the specific targeted data needs of the scientists, the MagnaVista application can be personalised in various ways. It currently supports user based selection of specific (remote) databases that should be interrogated; user based selection of the various data sets (fields) that should be returned from those databases; storage of specific genes of interest, as well as personalisation of the look and feel of the application itself.

The actual MagnaVista application itself is delivered to the users using Sun Web Start technology. Through launch buttons on the portal web page, a single mouse click can be used to automatically deliver the application and associated libraries, including the Web Start environment if it is not already present. However due to anomalies in Web Start with non-Internet Explorer versions of browsers used by the scientific community and issues of local firewalls blocking Web Start traffic, it was decided that a simpler version of this application was needed: GeneVista was produced to address these issues.

## 2.2 GeneVista

GeneVista is a portlet based application. Portlets are Java-based Web components, managed by a portlet container, that process requests and generate dynamic content. Portals use portlets as pluggable user interface components that provide a presentation layer to information systems. The next step, after servlets in Web application programming, portlets enable modular and user-centric Web applications.

In essence the functionality of GeneVista is very similar to MagnaVista. However, it does not support the richness of personalisation. We note that this was at the request of the scientific end users. They simply wanted to be able to select a collection of gene names and retrieve all available information. Few of them bothered with personalisation possibilities. The basic “Google-esque” front end to GeneVista was designed to reflect this. It is worth noting that the scientists predominantly use Google for their biological searching. Hence the look and feel of GeneVista has been designed to give them an environment and front end that they are comfortable with. GeneVista has however been designed to interrogate specific biological databases as opposed to gathering general information available across the internet as with Google.

The GeneVista portlet simply requires that the scientist input the gene names that they are interested in and selects submit. Following this, HTML based data sets are returned and presented within the browser window as shown in Figure 3.

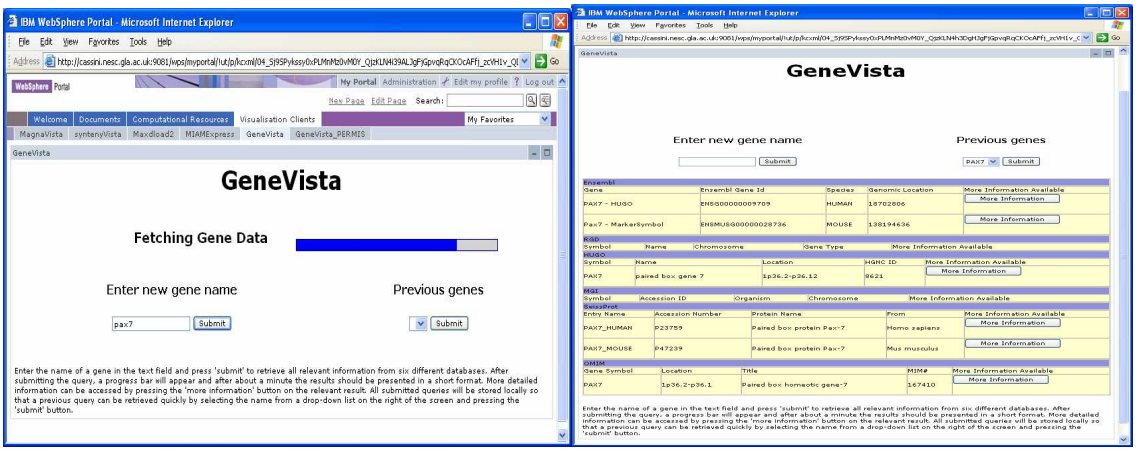


Figure 3: GeneVista Basic Usage for Gene Query

## 2.3 Lessons Learned in Developing Functional Genomic Data Grids

In developing this infrastructure, numerous lessons in developing and maintaining data Grid infrastructures for functional genomic data sets have been learnt. These include remoteness and independence of the data sets (including how they are independently managed); the data dependencies that exist between these remote data sets, as well as how technologies such as IBM Information Integrator and OGSA-DAI address these challenges.

### 2.3.1 Access to Remotely Managed Genomic Data

A major problem faced by both Information Integrator and OGSA-DAI is the changes made to the schema design associated with the remote data source. For BRIDGES, the two relational data sources which would allow public programmatic access were Ensembl [32] (MySQL - Rat, Mouse and Human Genomes, Homologs and Database Cross Referencing) and MGI [29] (Sybase - mainly Mouse publications and some QTL data.) Flat files were downloaded for Rat Genome Database [38] (RGD), OMIM [35] (Online Mendelian Inheritance in Man), Swissprot/Uniprot [37] (Protein sequences), HUGO [30] (Human Gene Ontology) and GO [44] (Gene Ontology). The other sites typically offer access to their data via front end HTML based web pages for example running CGI script; or they make their data in a flat file format on ftp servers without providing associated schemas to help in putting this into a local database.

Obviously changes made to the schema of a third part database are completely outwith the control of Grid infrtastructure developers. Ensembl change the name of their main gene database every month! The schema of the database has been drastically altered on at least 3 occasions during the BRIDGES project. MGI have had one major overhaul of all their table structure. In these cases queries to these remote data sources will fail. This is especially costly when large federated queries are made which ultimately fail when perhaps only one or two columns of a particular sub-query are problematic, i.e. are based on an erroneous schema.

The case of flat files is slightly different. Since the flat file has been downloaded onto the BRIDGES data server and either wrapped or parsed into the database, queries on this will survive but only until a newer schema-altered file is incorporated.

It should be noted that Information Integrator insists that the flat file being wrapped exists on a computer with exactly the same user setup and privileges as the data server itself. This is unlikely to be the case with regard to most curated life science data sets. It is also the case that the manual task of mapping flat file columns to database wrappers must still be done in the event of remote changes.

The ability to create Materialized Query Tables (MQTs) in Information Integrator can insulate the queries from these remote changes. An MQT is a local cache of a remote table or view and can be set to refresh after a specified time interval or not at all. Here we have a balancing act of deciding how important it is to have up to the minute data (refreshed frequently) or slightly older data but impervious to schema changes. The MQT can be optimized to try the remote connection first, and if it is unavailable to resort to the local cache but in the event that the remote connection is good but the query throws and exception because the schema has changed, then the query will fail.

BRIDGES has come up with a partial solution to the problem of remote schema changes. An application was developed (*Bridges_wget*) which systematically checks for database connections and if the connection is made runs a sample query specifically naming columns (so not a select *) to test if the table schema has changed. If all is well, remote flat files are checked for modification dates. If newer ones are found at the remote site they will be downloaded, parsed (if necessary) and either loaded into the database or the db administrator notified that new files are available.

Hopefully this will go some way to help in keeping the BRIDGES database up to date with current data. We note however that the parsers developed are not semantically intelligent so it would require updating the code (Java) to meet with file format modifications

### 2.3.2 Data Independence Issues

One of the issues around creating a federated view of remote data sources is the fact that these data sources are largely independent of each other. It is not always possible to find a column which will act as a foreign key over which the joining of the two (or more) databases can occur. When there is a candidate, often the column name is not named descriptively so to give a clue as to which database might be joined to.

Were all the databases developed by the same development team for example within a company intranet, this possibility of creating large scale joins across several homogenous databases would be much clearer. As it is one bio database may have a row with a gene identifier column with another column holding an accession ID for an entry for this gene in another database. In this way the join can be made. For example, in the case of Ensembl a row containing a gene identifier contains a Boolean column indicating whether a reference exists in another database. For example, RGD_BOOL=1 would indicate that a cross reference can be made to the RGD database for this gene identifier. We now have to query the Ensembl RGD_XREF table to obtain the unique ID for the entry in the RGD database. The query to RGD may then contain references to other databases and indeed back to Ensembl and one ends up with a circular referencing problem.

BRIDGES dealt with this problem by caching all the available unique identifiers along with the database in which it is found from all the remote data sources in a local materialized query table. When a match is found, the associated data resource is queried and all results returned to the user. It is then up to the user to decide which information to use.

In addition to the schema checking and file download programme (Bridges_wget), BRIDGES has developed a knowledge base for problems and solutions in an attempt at providing a project wide resource to assist developers. It can easily be extended to incorporate other projects and modules so that other projects working with DB2 for example can share their fixes and workarounds.

### 2.3.3 Data Return Size Issues

In practice with Information Integrator queries are issued from the MagnaVista or GeneVista application to stored procedures within DB2. Based on the parameters received the appropriate databases are queried. Rather than create one large join across all the remote data sources, with Information Integrator the stored procedure makes individual queries to each database and returns the result sets into a temporary table. It is this temporary table which is returned to the client.

With Information Integrator, selectively populating the temporary table allows us to make sure no duplication of data is returned. To illustrate this problem, an Ensembl gene identifier may be associated with several hundred publications in the MGI database and also a particular Swissprot accession ID and the taxon element is required to be returned from Swissprot. The taxon table is three relations away from the Accession table. There may be 5 taxons to be returned which means that there is no possibility of running a DISTINCT select statement. This would mean that all the publication data would be returned along with each taxon.

The fact that large publication data may well be involved in the bio database query could easily exceed the maximum tuple size returned from DB2. It is possible to increase the tuple size by increasing the page size for the database. This of course could work against performance if the bulk of the queries issued would return a small dataset and therefore there would be a redundancy in the page size overhead.

Using OGSA-DAI to perform the same type of query as has been outlined above is more problematic. To begin with there is no cache or local views available as with Information Integrator through MQTs. Instead there has to be a query made to each data resource to begin with in order to obtain all the unique identifiers. Of course these could be stored in a local file or database by doing a UNION of all identifiers on all databases thus serving as an alternative to the MQT.

The mapping of remote data sources to the ‘Integrator’ mechanism is always going to have to be done but with OGSA-DAI it is more a manual task rather than the automated ‘Discover’ function in Information Integrator.

There is a similarity in the way that the particular stored procedures were constructed within DB2 and the integration procedure in OGSA-DAI in that the former uses temporary tables to store the result sets from each database and the latter requires the existence of a database within which tables or temporary tables can be created to store each result set. It is from this newly constructed table that the client then obtains required data.

It should be noted that if MySQL is to be used as the 'dumping' database by OGSA-DAI, only newer releases support temporary tables and those that do are selective about allowing more than a single SELECT to be performed within a transaction on the temporary table. If actual tables were to be created there would have to be a convention in place regarding table naming otherwise collisions would occur.

The fact that stored procedures can be written in DB2 makes it simpler for the client as all the query processing is done within the body of the procedure but table-based queries can also be implemented in OGSA-DAI and the same results returned.

### 2.3.4 Issues in Creating Federated Views of Remote Data

When setting up a federated view in Information Integrator one first chooses a wrapper to use. Then one defines a 'Server' which contains all the connection parameters. Next 'Nicknames' are created for the server which are local DB2 tables mapped to their remote counterparts. To aid this process there is a 'Discover' function available when creating the Nicknames which when implemented will connect to the remote resource and display all the metadata available. One can then choose which of these to use, rename them and alter their schema for local use. Such advanced features are not available with OGSA-DAI.

It should be noted that if one drops any component, every component that is dependent on it will be dropped also. So for example, if a Server is dropped all Nicknames will be dropped. If a Nickname is dropped, all MQTs and Views based on that Nickname will also be dropped. Experience has shown that it is worth saving the original scripts.

We note that there is an unresolved issue regarding MQTs in Information Integrator. MQT's created with the same schema owner as the underlying tables from which they are built, are not accessible to users who do not have administrative privileges.

### 2.3.5 Enhancing the BRIDGES Data Grid Infrastructure

Whilst providing a simple mechanism through which a broad array of life science data sets can be simultaneously accessed and returned to the end user scientists, the richness of the functionality in both MagnaVista and GeneVista is somewhat restricted. Specifically, the queries issued to the DB2 database which are subsequently federated to the other databases returning results that are then joined are quite simple simplistic. They focus primarily on searching for gene names or synonyms. This provides a basis for application of Grid technologies for the CFG scientists, but does not provide them with the richness of vocabulary that they ideally need. Namely, to answer question such as "*has this microarray experiment been undertaken previously*" or "*how do my microarray results compare with general public knowledge and/or with the not yet formally published data sets at collaborating site X*".

To take these tools further, the GEMEPS project [52] is focusing upon access to and usage of microarray datasets at Cornell University Computational Biology Service Unit [21], the Riken Institute Japan [22] and the Sir Henry Wellcome Functional Genomics Facility. All of these sites have agreed to make available their microarray data sets and provide support in delivering Grid based security solutions. Thus rather than simply searching for gene names, we would like to be able to search for sets of genes that are associated with for example a particular microarray experiment. In addition, we would like to be able to extend this search capability with expression values and variations of expression values. Thus we plan to be able to issue queries of the form: *who has run an experiment which generated results including geneX, geneY and geneZ being expressed,* or further *who has generated results of the form geneX with expression value 75, geneY with expression value 66 and geneZ with expression value 23* (the numbers here are symbolic only and standard deviations will be needed), or queries of the form *who has generated results with expression values for geneX > geneY > geneZ.* In the first two cases much richer query capabilities than are supported

right now with our federated repository are needed. In the latter case, ordered sets of genes and their expressions may well suffice (given the variations of gene expressions in microarray experiments, this is likely to be a more realistic measure of the similarities of experiments). In this case, the query will be more aligned with solutions based around the Basic Local Alignment Search Tool (BLAST), since we are searching for patterns of genes ion microarray data warehouses.

In all of these cases, secure access to data sets at the remote sites is needed, since we want access to *live* experimental data and not just historical microarray results. Given the fact that this live research data is costly to produce and potentially contains important scientific results, a clear security policy is needed to define what data can be accessed and by whom, and under what circumstances. Thus if the results do lead to new insights, then appropriate attributions will reflect this. Similarly, the enforcement of this policy must be clearly satisfied by IT staff of the data owning site and hence by the remote scientists – site autonomy and security more generally is of paramount importance to the success of Grid computing.

## 3 Grid Based Security

With the open and collaborative nature of the Grid, ensuring that local security constraints are met and not weakened by Grid security solutions is paramount. Public Key Infrastructure (PKI) represents the most common way in which security is addressed. Through PKIs, it is possible to validate the identity of a given user requesting access to a given resource. For example, with the Globus toolkit [28] solution, gatekeepers are used to ensure that signed requests are valid, i.e. from known collaborators. When this is so, i.e. the Distinguished Name (DN) of the requestor is in a locally stored and managed gridmap file, then the user is typically given access to the locally set up account as defined in the file.

There are several key limitations with this approach with regard to security however. Most importantly, the level of granularity of security is limited. There is no mention of what the user is allowed to do once they have gained access to the resource. In principle we could launch any number of arbitrary processes. In the UK, this PKI works on the assumption that user certificates are provided by an acknowledged certificate authority (CA). To support this, a centrally managed CA at Rutherford Appleton Laboratories exists which (necessarily!) has strict procedures for how certificates are allocated. Users are expected to "prove" who they are in order to get a certificate, e.g. through presenting their passports to a trusted individual. This is a human intensive activity and one which has scalability issues once Grids are rolled out to the wider community such as industry and academia. Having users personally take care of their private keys is another limitation of this approach.

In short, current experiences with PKIs as the mechanism for ensuring security on the Grid have not been too successful [25,26]. Authorisation infrastructures, which allow expressing and enforcing what Grid users are allowed to do on a given Grid end-system, offer extended and finer grained security control when accessing and using Grid resources. The X.509 standard has standardised the certificates of a privilege management infrastructure (PMI). One of the leading authorisation infrastructures is from the Privilege and Role Management Infrastructure Standards Validation (PERMIS) project [8]. Through PERMIS, an alternative and more scalable approach to centrally allocated X.509 public key certificates can be achieved through issuing locally allocated X.509 authorisation based attribute certificates. These define the security policies that will be enforced when remote Grid based requests are received.

The PERMIS software realises a Role Based Access Control (RBAC) authorisation infrastructure. It offers a standards-based Java API that allows developers of resource gateways to enquire if a particular access to a resource should be allowed. The PERMIS RBAC system uses XML based policies defining rules, specifying which access control decisions are to be made for given VO resources. These rules include definitions of: subjects that can be assigned roles; SOAs ("Source of Authority" i.e. local managers) trusted to assign roles to subjects; roles and their hierarchical relationships; what roles can be assigned to which subjects by which SOAs; target resources, and the actions that can be applied to them; which roles are allowed to perform which actions on which targets, and the conditions under which access can be granted to roles. This Java API has been used within the BRIDGES project to define and subsequently enforce data security.

A key aspect of data security within BRIDGES has been to make it easy to use for the end users. The requirements and responsibilities that are incurred in acquiring and maintaining a UK e-Science certificate would have dissuaded many of the scientists from using the BRIDGES software. Hence the security infrastructure deployed had to be robust yet simple to use. We identified that the security infrastructure required to implement this should be separated into four components:

- A portal that provides access from remote clients onto a server. This allows a flexible, pervasive and highly available service to be presented to users across potentially large distances and crossing many domains of trust. This has been addressed using secure Hyper-Text Transport Protocol (https) to protect usernames and passwords during the transfer from the client machine to the server, through encryption.
- An authentication facility which verifies that users are who they say they are. This has been addressed using standard username and password pairs combined with the standard authentication mechanism provided by the portal software, IBM WebSphere. A secure user database within WebSphere stores user details.
- An authorization facility which determines what users can do based on their identity. We note that hard-coding of authorisation policies, rules and regulations within applications will not scale and is inflexible. A better approach is to authorise roles instead of individual users and have local policies defined and securely stored accessible to numerous applications which can then automatically check whether a given user is authorised to invoke the service in the manner they are attempting. This is achieved through the use of PERMIS [8]. PERMIS is a collection of applications for the definition of and enforcement of security policies. PERMIS integrates with Security Assertion Markup Language - the language used to implement the callout made by Globus services when providing authorization.
- An additional infrastructure that provides user management, auditing and logging facilities. This is achieved using a secure directory on the PERMIS authorization server that holds all attribute certificates relating to users. Detailed logging of all user actions has also been implemented.

The overall security infrastructure is depicted in Figure 4.

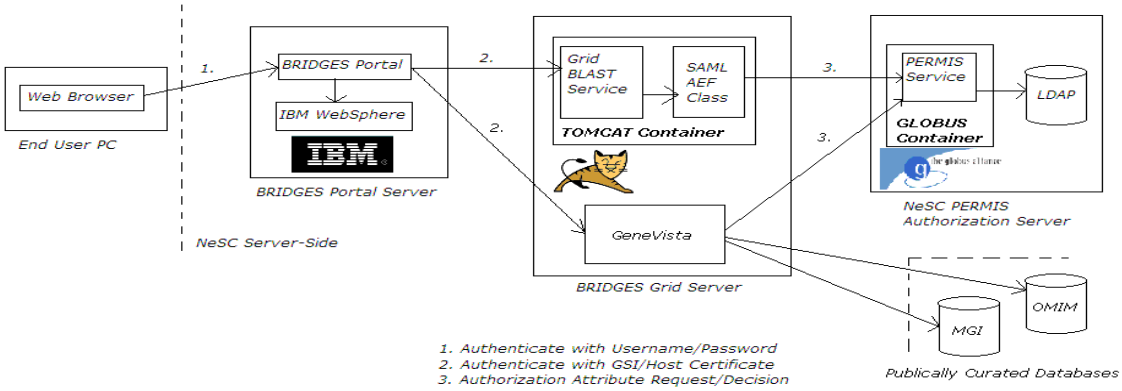


Figure 4: BRIDGES Security Infrastructure

With regard to data security, PERMIS policies have been defined and implemented restricting access to certain databases offered via the Data Hub to certain users. This is achieved through extensions to the GeneVista software to support queries of the PERMIS based LDAP policies. These policies distinguish CFG users from other users of the BRIDGES software. Specifically, the policies allow CFG scientists access to *all* of the data sets that are accessible from the federated repository. Other non-CFG users are allowed to create accounts on the portal, however they are only entitled access to the remote (federated) data sets accessible via the portal. It is important to note that this is all completely transparent to the end users when using GeneVista. They issue queries and receive results without any knowledge that a security policy has been enforced and that they are potentially only seeing a subset of the complete data sets depending on their role.

We note here that due to limitations in the security standards and associated implementations, the security policies are fixed and at the database level and not database contents level. For example, it has been recognised that the Grid security models for fine grained database access, e.g. to individual tables, rows or columns contained within databases is not easily supported in a scalable manner with the existing security infrastructures and standards. This is primarily due to the lack of support for parameters in the current Global Grid Forum Security Assertion Markup Language AuthZ interface [7]. Thus currently security can only be made on a method name (or database stored procedure that issues fixed SQL queries). A finer grained model where authorisation is based on the method name and associated parameters (or SQL *select* query with appropriate parameters/constraints say) is something that would be needed across many security-oriented projects. In a recent funded JISC project [53] Prof Chadwick at the University of Kent will be defining and implementing the enhanced Grid-authorisation infrastructure specification together with the Globus developers. We expect to explore these extensions to Grid security infrastructures in the course of the GEMEPS project and numerous other e-Health related projects at NeSC Glasgow.

## 4. Conclusions and Future Work

One of the major challenges facing in-silico life science research is managing the data deluge. Understanding biological processes necessitates access to and understanding of collections of potentially distributed, separately owned and managed biological data sets. Federated data models represent the most realistic approach due to the expense of centrally based data curation and the general reluctance of biologists to hand over their data to another party. Given that the CFG project are primarily interested in functional genomic based data resources, e.g. in supporting their microarray experiments, a bioinformatics workbench that allows the scientists to take up/down regulated gene names from microarray experiments and garner further information are of special interest. We note that the data sets accessible via the Data Hub are not a fixed set. Other resources can easily be added. However one of the challenges in this area is the issue in gaining access to these remote resources. For example, few of these resources provide the necessary programmatic access needed, i.e. to query their database directly. Instead, they often only offer compressed files that can be downloaded or made available via web front end solutions such as CGI scripts. As a result, the Data Hub includes a local warehouse for downloaded data sets. Currently federated access to ensembl (rat, mouse, human) and MGI is supported, with the other data sets having to be locally warehoused. This then requires that local (downloaded) data sets are kept up to date with the remote data sets. It is often non-trivial to establish a local repository that uses theses data sets, even with the local downloading of the data. Data providers often do not provide schemas or data models for the data themselves. Rather they simply provide large compressed text files that can be ftp'ed. A significant amount of effort is needed to understand how this data can be input into a database.

It is clear that many of the issues faced by all technologies is accessing and using life science data sets are impacted by standards and data models. Changing schemas is indicative of this. We are currently finalising a report on these issues funded by the MRC, BBSRC, JISC, NERC, DTI and Wellcome Trust [6] to be released imminently which outlines the political, technical, economic, social factors associated with data sharing.

It is clear that the microarray data producing and consuming community urgently require technology that will allow up to date microarray data information to be found, accessed and delivered in a secure, robust framework. We note that technological frameworks that allow to overcome the problems of dealing with a multitude of remote data sets existing on numerous end systems with different local security infrastructures is fundamental to much life science research. Grid based data access and integration technologies such as OGSA-DAIT, [33] in conjunction with appropriate standardisation efforts such as MIAME [4], MAGE-ML [39], MGED-OWG [40] for describing microarray experiments offer a viable data sharing framework. We believe the BRIDGES software provides a good basis upon which advanced data mining services dealing with remote data sets can be achieved.

The queries that are supported currently within BRIDGES are fairly simplistic in nature – returning all (or a subset) of the data sets associated with a named gene. Through the GEMEPS project we are now looking towards more complex queries, e.g. lists of genes that have been expressed and their up/down expression values as might arise in microarray experiments.

### 4.1 Acknowledgements

This work was supported by a grant from the Department of Trade and Industry. The authors would also like to thank members of the BRIDGES and CFG team including Prof. David Gilbert, Prof Malcolm Atkinson, Dr Dave Berry, Dr Ela Hunt and Dr Neil Hanlon. Magnus Ferrier is acknowledged for his contribution to the MagnaVista software, Dr Jos Koetsier for his work on the GeneVista application, Micha Bayer for feedback on earlier versions of this paper and work on portals and Grid based technologies within BRIDGES, Anthony Stell for his involvement on the security aspects of this work, and Derek Houghton for his work in developing the data repository. Acknowledgements are also given to the IBM collaborators on BRIDGES including Dr Andy Knox, Dr Colin Henderson and Dr David White. The CFG project is supported by a grant from the Wellcome Trust foundation.

## 5. References

[1] Human Proteome Organisation (HUPO), Proteomics Standards Initiative (PSI), http://psidev.sourceforge.net/
[2] Proteomics Experiment Data Repository (PEDRO), http://pedro.man.ac.uk/
[3] Architecture for Metabolomics (ArMet), www.armet.org
[4] Minimal Information About a Microarray Experiment (MIAME), http://www.mged.org/Workgroups/MIAME/miame.html
[5] Clinical Sciences Centre/Imperial College, http://microarray.csc.mrc.ac.uk/
[6] Joint Data Standards Survey (JDSS), http://www.d-archiving.com/JDSS/study.html
[7] Global Grid Forum, Frameworks and Mechanisms WG https://forge.gridforum.org/projects/authz-wg
[8] D.W.Chadwick, A. Otenko "The PERMIS X.509 Role Based Privilege Management Infrastructure". Future Generation Computer Systems, 936 (2002) 1–13, December 2002. Elsevier Science BV.
[9] L Pearlman, et al., A Community Authorisation Service for Group Collaboration, in Proceedings of the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks. 2002.
[10] Lepro, R., Cardea: Dynamic Access Control in Distributed Systems, NASA Technical Report NAS-03-020, November 2003
[11] Globus Grid Security Infrastructure (GSI), http://www-unix.globus.org/toolkit/docs/3.2/gsi/index.html
[12] Johnston, W., Mudumbai, S., Thompson, M. Authorization and Attribute Certificates for Widely Distributed Access Control, IEEE 7th Int. Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, Stanford, CA, June, 1998, p340-345 (http://www-itg.lbl.gov/security/Akenti/)
[13] Steven Newhouse, Virtual Organisation Management, The London E-Science centre, http://www.lesc.ic.ac.uk/projects/oscar-g.html
[14] BioMedical Research Informatics Delivered by Grid Enabled Services project (BRIDGES), www.nesc.ac.uk/hub/projects/bridges
[15] Dynamic Virtual Organisations in e-Science Education project (DyVOSE), www.nesc.ac.uk/hub/projects/dyvose
[16] Virtual Organisations for Trials and Epidemiological Studies (VOTES), www.nesc.ac.uk/hub/projects/votes
[17] Genomics and Healthcare Initiative (GHI), www.nesc.ac.uk/hub/projects/ghi
[18] Cardiovasular Functional Genomics project, www.brc.dcs.gla.ac.uk/projects/cfg
[19] MIAMexpress, www.ebi.ac.uk/miamexpress/
[20] MaxDLoad http://bioinf.man.ac.uk/microarray/maxd/maxdLoad/
[21] Computational Biology Service Unit, Cornell University, Ithaca, New York, http://www.tc.cornell.edu/Research/CBSU/
[22] Riken Genomic Sciences Centre Bioinformatics Group, Yokohama Institute, Yokohama, Japan http://big.gsc.riken.jp/
[23] Functional Genomics of NutriensTransport in Arabidopsis; Bioinformatics Approach, BBSRC grant, April 2003.
[24] Golub, T.R, et al, Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, www.sciencemag.org, Science, Vol 286, 15 October 1999.
[25] JISC Authentication, Authorisation and Accounting (AAA) Programme Technologies for Information Environment Security (TIES), http://www.edina.ac.uk/projects/ties/ties_23-9.pdf.

[26] Whitten, A., and Tygar, J. D. Why Johnny can't encrypt: a usability evaluation of PGP 5.0. Paper presented at the 9th USENIX security symposium, Washington, 1999.
[27] National Grid Service www.ngs.ac.uk
[28] Globus Toolkit http://www-unix.globus.org/toolkit/
[29] Mouse Genome Informatics (MGI), www.informatics.jax.org/
[30] US National Library of Medicine, http://www.nlm.nih.gov/
[31] PubMed Central Home, http://www.pubmedcentral.nih.gov/
[32] EMBL-EBI European Bioinformatics Institute, http://www.ebi.ac.uk/ensembl/
[33] Open Grid Service Architecture – Data Access and Integration Two (OGSA-DAIT), www.ogsadai.org
[34] IBM Information Integrator, http://www-306.ibm.com/software/data/
[35] NCBI Online Mendelian Inheritance in Man, http://www.ncbi.nlm.nih.gov/OMIM/
[36] El-Din El-Assal et al. A QTL for flowering time in *Arabidopsis* reveals a novel allele of CRY2.; Nature Genetics; 435-440, 2001.
[37] UniProt/Swiss-Prot, http://www.ebi.ac.uk/swissprot/
[38] Rat Genome Database (RGD), http://rgd.mcw.edu/
[39] MicroArray and Gene Expression Markup Language (MAGE-ML), http://www.mged.org/Workgroups/MAGE/mage-ml.html
[40] Microarray Gene Expression Data Society – Ontology Working Group, http://mged.sourceforge.net/ontologies/MGEDontology.php
[41] National Digital Curation Centre (DCC), www.dcc.ac.uk
[42] Scottish Bioinformatics Research Network (SBRN), www.nesc.ac.uk/hub/projects/sbrn
[43] Lightweight Directory Access Protocol (LDAP), www.openldap.org
[44] Gene Ontology (GO), http://www.ebi.ac.uk/GO/
[45] ScotGrid, www.scotgrid.ac.uk
[46] Brown, L. 2D-DIGE – Differential Gel Electrophoresis. ABRF2004 Annual Meeting: Integrating Technologies in Proteomics and Genomics, Portland, Oregon, 28 Feb-2 March, 2004.
[47] Gygi, S.P., Quantitative Analysis of Complex Protein Mixtures using Isotope-coded Affinity Tags, Journal of Nature Biotechnology vol 17, October 1999 http://biotech.nature.com
[48] Barrow, M.P., Principles of Fourier Transform Ion Cyclotron Resonance Mass Spectrometry and its Application in Structural Biology, Analyst Journal, 2005, Vol 130 (1), pages 18 – 28.
[49] Herzyk, P., et al. A simple, yet powerful, new method to detect differentially regulated genes in replicated microarray experiments, FEBS Lett. 573:83-92, 2004.
[50] Herzyk, P., et al, Iterative Group Analysis (iGA): A simple tool to enhance sensitivity and facilitate interpretation of microarray experiments, BMC Bioinformatics, 5:34, 2004.
[51] Herzyk, P., et al, Graph-based iterative Group Analysis enhances microarray interpretation, BMC Bioinformatics, 5:100, 2004.
[52] Grid Enabled Microarray Expression Profile Search project (GEMEPS), BBSRC grant, www.nesc.ac.uk/hub/projects/gemeps
[53] Prof David Chadwick, Authorisation Interface V2 for the Grid, June 2005, JISC proposal accepted for funding.